المدرسة الوطنية العليا في الأمن السيبراني
NATIONAL SCHOOL OF CYBERSECURITY

1ST YEAR BASIC TRAINING IN CYBER SECURITY

INTRODUCTION TO OPERATING SYSTEMS 1 (SYST1)

Dr. Sassi BENTRAD
✉ : sassi.bentrad.enscs@gmail.com | | sassi.bentrad@enscs.edu.dz
LISCO Laboratory (**L**aboratoire d'**I**ngénierie des **S**ystèmes **CO**mplexes) / University of Badji Mokhtar-Annaba (**UBMA**)
National **S**chool of **C**yber **S**ecurity (**NSCS**)

Basic Training in Cyber Security (1BT)

Formation de Base en Cyber-Sécurité (1FB)

Chapter 3

Linux Key Commands

SYST1'2025/2026

LINUX KEY COMMANDS (10 %)

COURSE CONTENT

CHAPTER 4

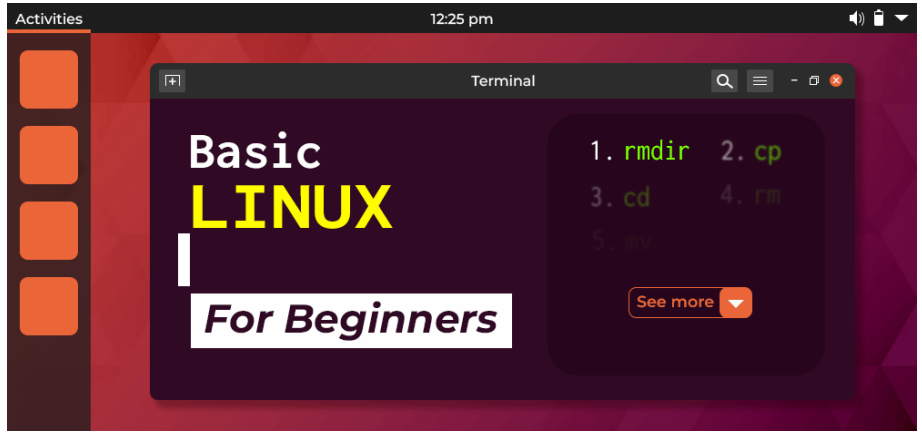
- Introduction
- Command Syntax
- Basic Commands
- Command's Documentation
- Regular Expressions
- Advanced Commands

❑ Introduction ...

Goal :
Discover the **essential Linux commands** for navigating and managing your system through the **terminal environment**.

What you'll Learn :

- 🧩 Understand **command syntax and structure**
- 📘 Explore **command documentation** tools (e.g., man, help)
- 🔍 Apply **regular expressions** for text search and filtering
- ⚙️ Experiment with **advanced commands** for automation and system control

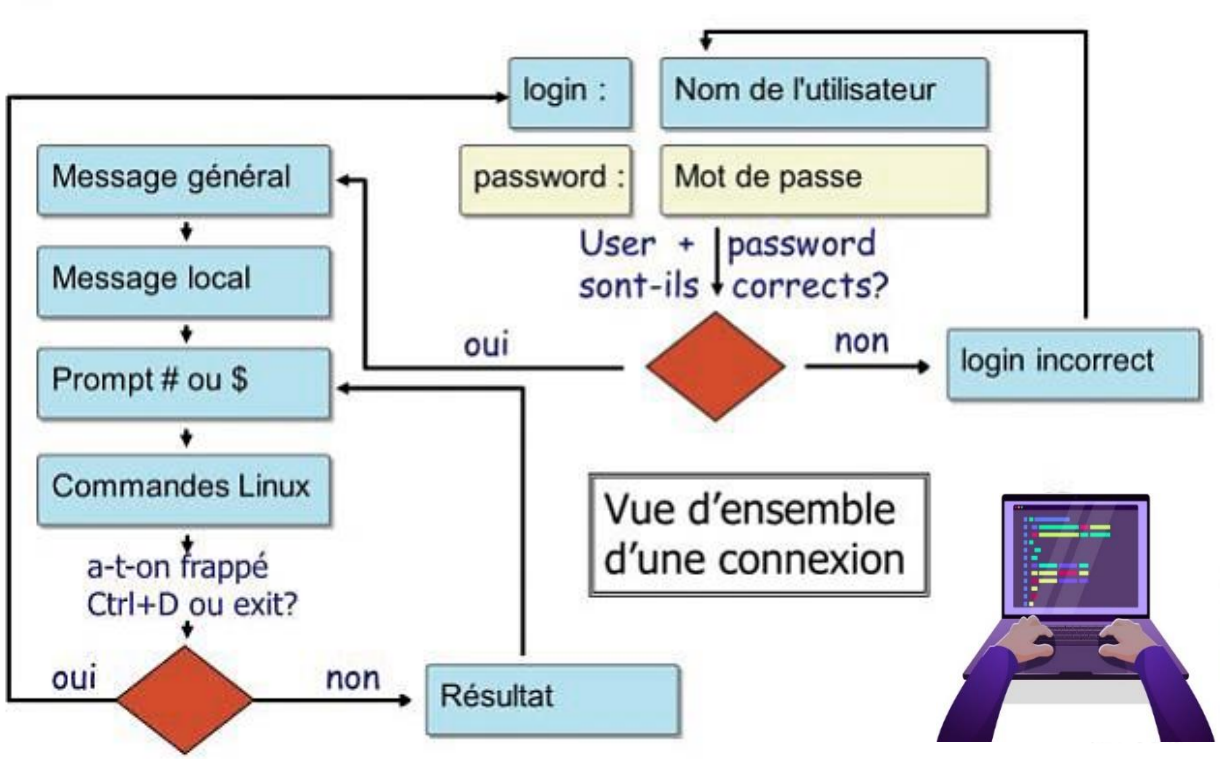


By the end of this Chapter :
You'll have a solid foundation to **confidently interact with any Linux environment** using the **C**ommand **L**ine **I**nterface (**CLI**).

❑ Commands ... ?

- **Commands** are used in both **graphical mode** and in **command line** (**console**) **mode**.
- **Commands** are entered into the **console** to ask the computer to perform actions.
- The **console** displays a **Command Prompt** at the beginning of the line ('**$**' or '**#**' or '**%**'), to tell the user that they are waiting for a command to be entered.
- This **prompt** reminds you of your username, the name of the machine, and the folder you are in.

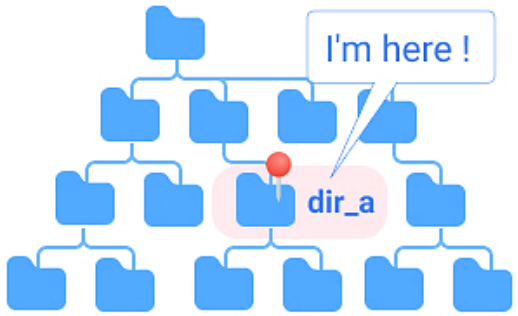
❑ How the SHELL works ?



❑ Use of Linux Commands

- A **Linux command** is an action to be performed or a program to be started.
- **Linux commands** are the **basic tools** used to interact with **Linux environment** on an individual level.
- **Linux commands** are used to perform a variety of tasks.
- The **command-line mode** allows for more complex operations.

Directory Tree and Current Working Directory on GUI



Current Working Directory on CUI

❑ What exactly are Commands ?

A command can be one of **four different things**.

- **An executable program** like all those files we saw in ***/usr/bin***. Within this category, programs can be ***compiled binaries*** such as programs written in C and C++, or programs written in *scripting languages* such as the shell, Perl, Python, Ruby, and so on.

- **A command built into the shell itself**. bash supports a number of commands internally called ***shell built-ins***. The command **cd**, for example, is a ***shell built-in***.

- **A shell function**. Shell functions are miniature **shell scripts** incorporated into the *environment*. We will cover configuring the environment and writing shell functions in later chapters, but for now, just be aware that they exist.

- **An alias**. Aliases are commands that we can define ourselves, built from other commands.

❑ Command Syntax

- A **Linux command** is broken down into **three parts**:

COMMAND	[OPTIONS]	[AGRGUMENTS]
Command-name	**[options]**	**[parameters]**

  - ✓ **Command-name** : The first word corresponds to the command.
  - ✓ **Options**: Introduced by a ***hyphen*** "**-**", allows you to change the behavior of the command (optional).
  - ✓ **Arguments**: The objects or files on which the command acts.

- To run it, simply type its name followed by options and arguments.
- After entering the first letters of a command, you can complete its name using the **Tab key**.
- Previous commands can be found using the **Up** and **Down** arrows, or search through previous commands with **Ctrl + R**.
- **There are many other keyboard shortcuts** that are recommended to know in order to be able to enjoy the console to the fullest.

Command Syntax

COMMAND	[OPTIONS]	[AGRGUMENTS]
Command-name	**[options]**	**[parameters]**

Short options (one letter):

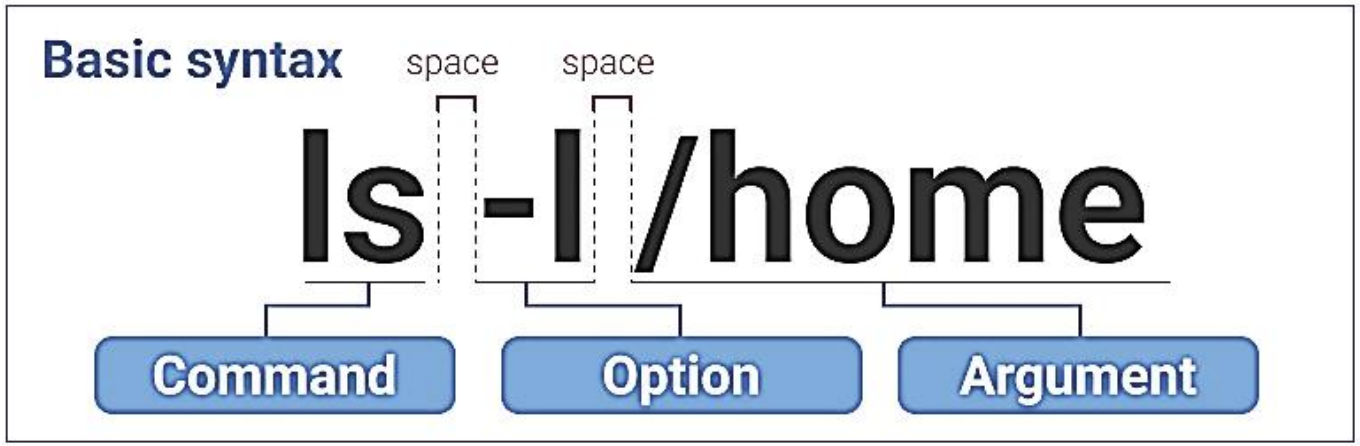
- ✓ A letter preceded by a **hyphen** (–) : **ls –a**
- ✓ Multiple options : **ls –a –h –l** or **ls –ahl**

Long options (several letters):

- ✓ Options followed by **two hyphens** (– –) : **ls – –all**
- ✓ Multiple options : **ls –hls – –all**

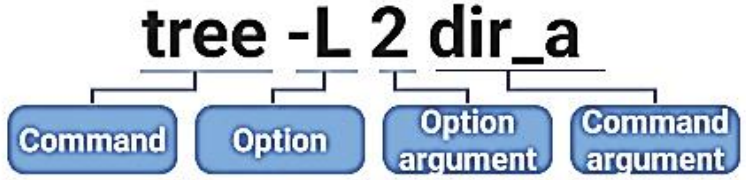
❑ Command Syntax

COMMAND	[OPTIONS]	[AGRGUMENTS]
Command-name	**[options]**	**[parameters]**



With multiple arguments

With option argument



Command Syntax

COMMAND	[OPTIONS]	[AGRGUMENTS]
Command-name	[options]	[parameters]

```
# command without options and parameters
user@PC:~$ ls

# command with options
user@PC:~$ ls -a -t

# command with parameters
user@PC:~$ ls Desktop/ Documents/

# command with options and parameters
user@PC:~$ ls -a -t Desktop/ Documents/
```

❑ Command Syntax

[xx] argument

ls (List Contents of Directory)

`ls [directory path]`

No option: list directories and files

```
application  media  sample.txt
```

-a option: list with hidden files

```
.hidden_file application  media  sample.txt
```

-l option: list with detailed information

```
drwxr-xr-x   3 usr_a staff  4096 Jul 16  2020 media
-rwxr-xr-x  15 usr_b staff  3180 Dec 21 12:53 sample.txt
```

1. directory or file (d = directory)
2. access modes
3. # of hardlinks
4. owner
5. group
6. size (bytes)
7. modification date and time
8. name of file or directory

❑ Identifying Commands

☒ type — *Display a Command's Type*

There are **two types of commands** :

- ✓ **Internal** shell commands (part of shell program)
- ✓ **External** commands (so we'll receive the **location of the command**)

☒ which — *Display an Executable's Location*

```
# internal command
user@PC:~$ type history
history is a shell built-in

user@PC:~$ type pwd
pwd is a shell built-in

# external command
user@PC:~$ type hostname
hostname is /bin/hostname

user@PC:~$ type date
date is /bin/date
```

❑ Examples : *Gathering System Information using the Linux Shell*

Objective: Use various shell commands to gather and display various pieces of **information about your system**, such as hardware and software details, system performance, disk usage, memory, and more.

Tasks:

Basic System Information :

- Find the **hostname** of the system.
- Determine the **Linux kernel version** you are running.
- Display the **current logged-in users**.

Hardware Information :

- Find out the **CPU model** and number of cores.
- Check the **total RAM** available on your system.
- Get the **disk space usage** for all mounted filesystems.
- Display the **PCI devices** connected to your system.

System Uptime and Load :

- Find out the **system uptime** (how long the system has been running).
- Display the **system load averages** (1, 5, and 15 minutes).

❑ Examples : *Gathering System Information using the Linux Shell*

Objective: Use various shell commands to gather and display various pieces of **information about your system**, such as hardware and software details, system performance, disk usage, memory, and more.

Tasks:

Network Information :

- Display the **IP address** assigned to your system.
- List all active **network interfaces** on the system.
- Show the **current network connections**.

File System Information :

- Display detailed information about the **file system type** and **mounted file systems**.
- Show the **total disk space** and **available space** on the root partition.

Process and Resource Usage :

- Show a list of the **top processes** consuming the most CPU.
- Display the **top processes** consuming the most memory.
- Display the **system's current memory usage** (used, free, and available).

Examples : *Gathering System Information using the Linux Shell*

Commands to Use :

Basic System Information:

- Hostname: **hostname**
- Kernel Version: **uname -r**
- Logged-in Users: **who** or **w**

Hardware Information:

- CPU Model & Cores: **lscpu** or **cat /proc/cpuinfo**
- RAM: **free -h** or **cat /proc/meminfo**
- Disk Usage: **df -h**
- PCI Devices: **lspci**

System Uptime and Load:

- Uptime: **uptime**
- Load Average: **uptime** or **top** (look at the first line)

Network Information:

- IP Address: **hostname -I** or **ip a**
- Network Interfaces: **ip link** or **ifconfig**
- Active Network Connections: **ss -tuln** or **netstat -tuln**

File System Information:

- File System Type: **df -T**
- Disk Space: **df -h /**

Process and Resource Usage:

- Top 5 Processes by CPU: **ps aux --sort=-%cpu | head -n 6**
- Top 5 Processes by Memory: **ps aux --sort=-%mem | head -n 6**
- Memory Usage: **free -h**

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

- Abbreviation of **man**ual pages.

  `man <command>` *(Display the manual pages of the command ...)* **Example: man ls**

- **Searching within man :**

  To search all available manual entries for a **keyword** (*pages that relate to a keyword*), use :

  ```
  man -k < keyword >
  ```

  Example : `man -k bash`

- How to use the **manual** : ... ?

  - ✓ Keys ↑ ↓ to move line by line.
  - ✓ **Home** key to go to the beginning and **End** key to go to the end.
  - ✓ **PageUp** and **PageDwon** keys to move page by page.
  - ✓ **/** key to search.
  - ✓ Press **q** to exit.

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

- Change the language of the manual:

✓ Ability to change the language of the manual.

✓ Install French: **apt –get install manpages-fr**

✓ Uninstall French: **apt –get autoremove mangages-fr**

✓ However, the pages are **not always up to date.**

✓ Work with the pages of the manual in **English**.

Manual page files are looked for in the directories specified by the **MANPATH** environment variable.

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

Structure of a Manual Page :

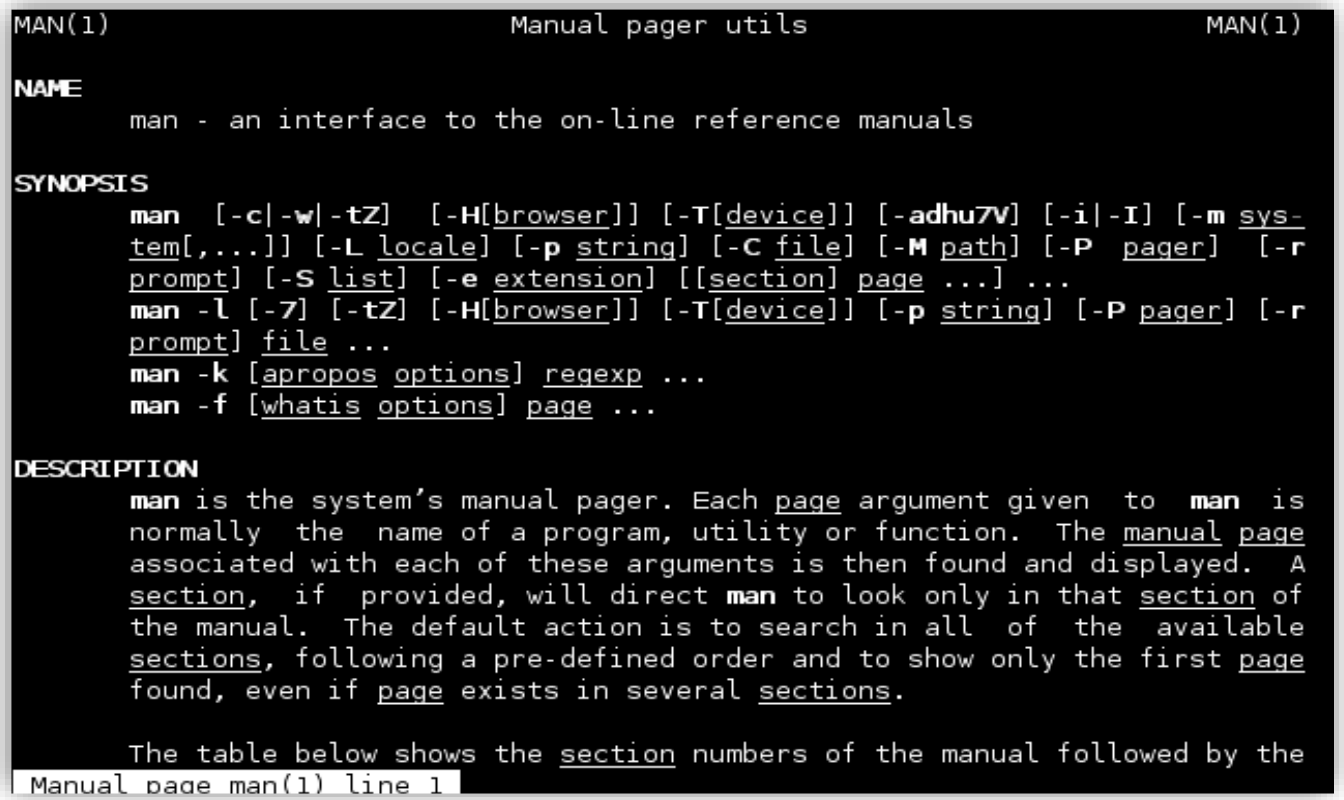
All the manual pages are organized, in the same way, into the following sections:

1. **NAME** - The name of the command or function and a short description.
2. **SYNOPSIS** - Syntax and options required to use the command.
3. **DESCRIPTION** - Details on what the command does, often with more explanation of options.
4. **OPTIONS** - List of flags and arguments, with explanations.
5. **EXIT STATUS** - Information about the exit status codes.
6. **EXAMPLES** - Usage examples, though not all man pages include examples.
7. **SEE ALSO** - References to related commands or topics.
8. **BUGS** - Known bugs or issues.

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

Navigating Manual Pages :

- **Move down one line:** Down Arrow or j
- **Move up one line:** Up Arrow or k
- **Scroll down one page:** Spacebar
- **Scroll up one page:** b
- **Search:** / keyword and press 'n' to jump to the next occurrence of the search term, and 'p' moves to the previous occurrence.
- **Quit :** the 'q' key



❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

Manual Page Organization :

Man pages are divided into **8 numbered sections**, which group similar types of documentation.

- **(1)** User commands (e.g., **ls**, **bash**)
- **(2)** System calls
- **(3)** Library calls
- **(4)** Special files (e.g., files in **/dev**)
- **(5)** File formats and conventions
- **(6)** Games and screensavers
- **(7)** Miscellaneous (e.g., macros, conventions)
- **(8)** System administration commands (e.g., **ifconfig**)

Section	Contents
1	User commands
2	Programming interfaces for kernel system calls
3	Programming interfaces to the C library
4	Special files such as device nodes and drivers
5	File formats
6	Games and amusements such as screen savers
7	Miscellaneous
8	System administration commands

Viewing a page of the manual : `man [-s section] < command >`

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

SYNOPSIS Section :

```
SYNOPSIS
       ls [OPTION]... [FILE]...
```

- ls: To use the command **ls**, type **ls** firstly.
- [OPTION] : After the command, it is possible to write one or more options.
- [FILE] : The name of the directory for which to list the information.
- **...** : after OPTION, possibility to put several options. After FILE, it is possible to specify several directories.
- [] : facultative
- **bold** : words to be typed literally.
- **Underlined** : words to be replaced by the appropriate one.

The **options** are listed in the section **"DESCRIPTION"** of the manual page.

❑ Getting a Command's Documentation : man — *Display a Program's Manual Page*

SYNOPSIS Section :

```
SYNOPSIS
       mv [OPTION]... [-T] SOURCE DEST
       mv [OPTION]... SOURCE... DIRECTORY
       mv [OPTION]... -t DIRECTORY SOURCE...
```

Example: `man mv`

- **mv** [OPTION] **…** [-T] SOURCE CIBLE
  Rename the SOURCE (name of the file to be renamed) to TARGET (the new name CIBLE).

- **mv** [OPTION] **…** SOURCE **…** DIRECTORY
  Move the source to the DIRECTORY (one or more items can be moved).

- **mv** [OPTION] **…** [-t] DIRECTORY SOURCE **…**
  Move all SOURCE items to the DIRECTORY.

❑ Getting a Command's Documentation : help — *Get Help for Shell Built-in*

- **Getting Help Quickly :** If man pages feel too detailed, many commands also have **`--help`** or **`-h`** options to display a brief usage summary.
- Easier to read summarized help display.
- Use in addition to the command **man**
- Some commands do not have a manual pages, but the possibility of using **`-h`** (sometimes **`--help`**).

Example : **`cd --help`**

```
:~$ cd --help
cd: cd [-L|[-P [-e]] [-@]] [rép]
    Change le répertoire de travail du shell.

    Change le répertoire actuel vers DIR.  Le répertoire DIR par défaut
    est donné par la variable « HOME » du shell.

    La variable CDPATH définit le chemin de recherche du répertoire contenant
    DIR. Les noms de répertoires alternatifs dans CDPATH sont séparés par un deux-point « : ».
    Un nom de répertoire vide est identique au répertoire actuel.  Si DIR commence
    avec une barre oblique « / », alors CDPATH n'est pas utilisé.

    Si le répertoire n'est pas trouvé et que l'option « cdable_vars » du shell est définie,
    alors le mot est supposé être un nom de variable. Si la variable possède une valeur,
    alors cette valeur est utilisée pour DIR.

    Options :
      -L        force le suivi des liens symboliques : résout les liens symboliques dans
                DIR après le traitement des instances de « .. »
      -P        utilise la structure physique des répertoires sans suivre les liens
                symboliques : résout les liens symboliques dans DIR avant le traitement des
                instances de « .. »
      -e        si l'option -P est fournie et que le répertoire de travail actuel ne peut pas
                être déterminé avec succès, alors sort avec un code de retour non nul
      -@  sur les systèmes qui le supporte, présente un fichier avec des attributs
                étendus comme un répertoire contenant les attributs du fichier
```

❑ Getting a Command's Documentation: apropos — *Display Appropriate Commands*

- Search for an unknown command.
- **Keyword** as a parameter that will be searched in the pages of the manual.
- The command **apropos** allows you to find a command.

Example: Search for a command related to ***directories***.

```
:~$ apropos directories
addgnupghome (8)      - Create .gnupg home directories
cp (1)                - copy files and directories
dh_bugfiles (1)       - install bug reporting customization files into package...
dh_clean (1)          - clean up package build directories
dh_compress (1)       - compress files and fix symlinks in package build direc...
dh_fixperms (1)       - fix permissions of files in package build directories
dh_install (1)        - install files into package build directories
dh_installchangelogs (1) - install changelogs into package build directories
dh_installdebconf (1) - install files used by debconf in package build direct...
dh_installdirs (1)    - create subdirectories in package build directories
dh_installdocs (1)    - install documentation into package build directories
dh_installexamples (1) - install example files into package build directories
dh_installinit (1)    - install service init files into package build directories
dh_installman (1)     - install man pages into package build directories
dh_installmenu (1)    - install Debian menu files into package build directories
dh_installmime (1)    - install mime files into package build directories
dh_link (1)           - create symlinks in package build directories
dh_lintian (1)        - install lintian override files into package build dire...
dh_usrlocal (1)       - migrate usr/local directories to maintainer scripts
fc-scan (1)           - scan font files or directories
```

❑ Getting a Command's Documentation:

whatis — *Display One-line Manual Page Descriptions*

- The command **whatis** displays the name and a one-line description of a **man page matching a specified keyword.**
- It is intended to provide **a concise explanation** of the utility of the command in question.
- Returns the manual header (NAME).

Example : `whatis cp`

```
:~$ whatis cp
cp (1)                   - copy files and directories
```

❑ Getting a Command's Documentation : info — *Display a Program's Info Entry*

- In **GNU**, **man pages** are being replaced by **info pages**.
- Some manual pages even tell to refer to info pages instead.

```
info < command >
```

Features of info pages :

✓ Documentation structured in **sections** ("*nodes*") and **subsections** ("*subnodes*")

✓ Possibility to navigate in this structure: *top*, *next*, *prev*, *up*

✓ **Info pages** generated from the same text info source as the HTML documentation pages

❑ Manual pages / Useful techniques for searching

In Linux, the **man** (manual) command is a **powerful tool** for accessing documentation about other **commands**, **functions**, or **files**. You can search through the man pages using ***patterns*** with **/** and ***regular expressions***.

By mastering these techniques, we can **efficiently navigate** and **extract information** from the **manual pages**.

1. Basic Search in Manual Pages

Example: `/your_pattern`

- ✓ While viewing a man page, type **/** followed by your **search term**, then press **Enter**. This will search forward from your current position in the document.
- ✓ To move to the next occurrence of the **pattern**, press **n**. To move to the previous occurrence, press **N**.

2. Case-Insensitive Search

Example: `/\cpattern`

- ✓ To make the search case-insensitive, you can **prefix** the search term with **\c**.

❑ Manual pages / Useful techniques for searching

3. Regular Expressions

You can use regular expressions within **man** to find **patterns**.

Examples:

- **Search for a line starting with a word**: `/^pattern`
- **Search for words ending with a specific suffix**: `/suffix$`
- **Search for any digit**: `/[0-9]`

4. Search only in Specific Sections

Linux **man pages** are divided into **sections** (e.g., *commands, system calls, library functions*). You can search only in a specific section by specifying it.

Example : `man 5 passwd`

This will search for the passwd **man** page in **section 5** (file formats), instead of the default.

❑ Manual pages / Useful techniques for searching

5. Search across Manual Pages with apropos or man -k

Use **apropos** or **man -k** to search for a term across all man pages.

Example : `apropos pattern` **or** `man -k pattern`

6. Combine with grep for more Complex Searches

You can combine **man** with **grep** to filter out lines matching a pattern.

Example: `man ls | grep -i "option"`

This will display only lines from `man ls` that contain "**option**," ignoring case.

❑ Manual pages / Useful techniques for searching

grep (Global Regular Expression Print)

[xx] argument

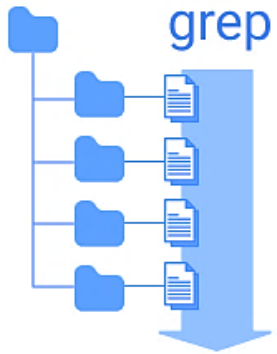
grep [*search pattern*] [*file or directory path*]

Phrase or regular expression

- Search for a string of characters (phrase) or a pattern in specified files
- Return the lines with the string

-r (recursive) option

search a string in multiple files under a specified directory



❑ Searching the Internet for resources (1)

Investigating issues :

- Most **forums** and **mailing list archives** are public, and are indexed on a very frequent basis by **Google**.

- If you investigate an error message, copy the message error in the search form, enclosed in double quotes ("error message").

  Lots of chances that somebody else already faced the same issue.

- Don't forget to use **Google Groups** : http://groups.google.com/

  This site indexes more than 20 years of newsgroups messages.

❑ Searching the Internet for resources (2)

Looking for documentation :

- Look for **<tool>** or **<tool> page** to find the tool or project home page and then find the latest documentation resources.
- Look for **<tool> documentation** or **<tool> manual** in your favorite search engine.

README and other Program Documentation Files

Looking for generic technical information :

- **WikiPedia** : http://wikipedia.org

  Lots of useful definitions in computer science. A real encyclopedia! Open to anyone's contributions.

❑ Advanced Commands ...

...

THANK YOU for your attention!

Questions ?

NSCS
المدرسة الوطنية العليا في الأمن السيبراني
NATIONAL SCHOOL OF CYBERSECURITY

مخبر هندسة الأنظمة المعقدة
LABORATOIRE D'INGÉNIERIE DES SYSTÈMES COMPLEXES

For more information about my research works, **Contact Information**:

Dr. Sassi BENTRAD

LISCO Laboratory : http://lisco.univ-annaba.dz/

☎ : +213 …

✉ : sassi.bentrad.enscs@gmail.com // sassi.bentrad@enscs.edu.dz

in : www.linkedin.com/in/sassi-bentrad/

: http://www.bentrad-sassi.sitew.com/

ORCID
Connecting Research and Researchers

: orcid.org/0000-0002-7458-8121

RESEARCHERID : A-9442-2013

SCOPUS Author ID : 44461052600